



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TJ 141502**

**PENGEMBANGAN *PLATFORM* PEMANTAUAN  
PERTUMBUHAN TERNAK PADA PETERNAKAN BERBASIS  
KOMPUTASI AWAN**

Nailul Fadloil  
NRP 2913 100 019

Dosen Pembimbing  
Prof. Dr. Ali GÜNEŞ  
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TJ 141502**

**PENGEMBANGAN *PLATFORM* PEMANTAUAN  
PERTUMBUHAN TERNAK PADA PETERNAKAN BERBASIS  
KOMPUTASI AWAN**

Nailul Fadloil  
NRP 2913 100 019

Dosen Pembimbing  
Prof. Dr. Ali GÜNEŞ  
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TJ 141502**

**DEVELOPMENT PLATFORM CATTLE'S GROWTH  
MONITORING ON FARM BASED ON CLOUD COMPUTING**

Nailul Fadloil  
NRP 2913 100 019

Advisor  
Prof. Dr. Ali GÜNEŞ  
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTMENT OF COMPUTER ENGINEERING  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Pengembangan Platform Pemantauan Pertumbuhan Ternak Pada Peternakan Berbasis Komputasi Awan”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

Nailul Fadloil  
NRP. 2913100019

## LEMBAR PENGESAHAN

### **Pengembangan Platform Pemantauan Pertumbuhan Ternak pada Peternakan Berbasis Komputasi Awan**

Merupakan hasil terjemahan dari buku *Graduation Project* yang berjudul :

***Development Platform Cattle's Growth Monitoring on Farm Based on  
Cloud Computing***

Oleh : Nailul Fadloil (NRP: 2913100019)

yang telah dipresentasikan di Departemen Teknik Komputer, Fakultas  
Teknik, Istanbul Aydin University pada tanggal 5 Juni 2017. Dengan dosen  
penguji :

1. Prof. Dr. Ali GÜNEŞ
2. Prof. Dr. Zafer ASLAN
3. Dr. Parvaneh SHAMS

Mengetahui  
Kepala Departemen Teknik Komputer

Dr. I Ketut Eddy Purnama, S.T., M.T.  
NIP. 196907301995121001

# ABSTRAK

Nama Mahasiswa : Nailul Fadloil  
Judul Tugas Akhir : Pengembangan *Platform* Pemantauan  
Pertumbuhan Ternak pada Peternakan  
Berbasis Komputasi Awan  
Pembimbing : 1. Prof. Dr. Ali GÜNES  
2. Dr. Supeno Mardi S. N., ST.,MT.

Pengelolaan peternakan yang ada di Indonesia rata-rata masih menggunakan cara konvensional. Adapun aplikasi yang dikembangkan untuk mengelola peternakan masih berupa aplikasi desktop yang tidak terintegrasi dengan jaringan. Aplikasi semacam itu tidak dapat digunakan untuk peternakan yang bersifat multiarea. Pengelola peternakan juga tidak dapat memantau peternakannya secara remote. Oleh karena itu butuh dikembangkan sebuah platform bisnis pemantuan peternakan berbasis komputasi awan. Dalam tugas akhir ini akan dikembangkan sebuah platform pemantuan peternakan dengan menggunakan komputasi awan sebagai pusat kegiatan perekaman, pengolahan dan analisa data. Komputasi yang dilakukan pada satu tempat memberikan kemudahan berupa integrasi data ketika melakukan pembaruan maupun perbaikan. Dengan menggunakan jaringan internet kegiatan-kegiatan tersebut dapat diakses secara remote dan realtime atau semi realtime. Sehingga dapat menghadirkan aplikasi pemantuan peternakan yang lebih efektif dan efisien bagi pengelola peternakan.

Kata kunci: Komputasi Awan, Sistem Manajemen Peternakan, Aplikasi Bergerak, Android

# Abstract

Student Name : Nailul Fadloit  
Project Title : Development Platform Cattles Growth  
Monitoring on Farm Based on Cloud  
Computing  
Advisors : 1. Prof. Dr. Ali GNE  
2. Dr. Supeno Mardi S. N.,ST.,MT.

*These days livestock farming can contribute to huge profit income to the farmers. One of problem in Indonesian livestock farming is manual data collection and analysis. Livestock management in Indonesia currently using traditional way which is not efficient. Some application developed for livestock management system still using desktop based which is not integrated by network. This kind of application is not fit to multi-area farming. Monitoring by remote cannot be reach by this system also. Besides, it need more cost because each farmer need to build and buy their application. Cloud computing technologies can be one of solutions to this problem through enhancing traditional livestock management system, by integrating processes, monitoring, operational farm management with cloud specific capabilities which will increase productivity and efficiency. Based on this idea I propose to develop a cloud based integrated intelligent system to improve efficiency in monitoring livestock condition. The cloud system provides information and data related to the farm and do analysis for each animal condition which will help farmers to take decision or supported the decision taking processes.*

*Keywords: Cloud Computing, Livestock Management System, Mobile Application, Android*

*Halaman ini sengaja dikosongkan*



# KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Pengembangan *Platform* Pemantauan Pertumbuhan Ternak pada Peternakan Berbasis Komputasi Awan**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, dan Ayah yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini
2. Bapak Kepala Departemen Teknik Komputer Dr. I Ketut Eddy Purnama, ST., MT.
3. Bapak Prof. Dr. Ali GÜNES, Bapak Dr. Supeno Mardi S. N., ST.,MT. dan Bapak Dr. I Ketut Eddy Purnama atas bimbingan selama mengerjakan penelitian
4. Bapak-ibu dosen pengajar Bidang Studi Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini
5. Seluruh teman-teman B401 Laboratorium Bidang Komputasi Multimedia, Aditya Very Cleverina, Gokmen GOKMEN, Berkay KANCA, Kayahan BARAKAZI, Elif OLCAY dan teman-teman saya selama mengikuti Program Pertukaran pelajar di *Istanbul Aydin University*

Kesempurnaan hanya milik Allah SWT, kritik dan saran pengembangan untuk penelitian ini sangatlah dibutuhkan demi terciptanya teknologi yang lebih baik.

Surabaya, 6 Juli 2017

Penulis

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>Abstrak</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>KATA PENGANTAR</b>	<b>ix</b>
<b>DAFTAR ISI</b>	<b>xi</b>
<b>DAFTAR GAMBAR</b>	<b>xiii</b>
<b>DAFTAR TABEL</b>	<b>xv</b>
<b>DAFTAR KODE</b>	<b>xvii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar belakang . . . . .	1
1.2 Permasalahan . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan masalah . . . . .	3
1.5 Sistematika Penulisan . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Komputasi Awan . . . . .	5
2.1.1 Manfaat Komputasi Awan . . . . .	6
2.2 Google Firebase . . . . .	7
2.3 Membuat Struktur Basisdata pada JSON Tree . . .	10
2.4 Library Android Pendukung . . . . .	13
2.4.1 Android Graph View . . . . .	13
2.4.2 Picasso . . . . .	13
<b>3 DESAIN SISTEM</b>	<b>15</b>
3.1 Desain Sistem . . . . .	15
3.1.1 Proses Pengumpulan Data . . . . .	16
3.1.2 Proses Pembacaan Data . . . . .	17
3.1.3 Proses Pemberitahuan . . . . .	17
3.2 Desain Aplikasi Berbasis UML . . . . .	18

3.3	Desain Antarmuka Aplikasi . . . . .	21
3.4	Desain Basis Data . . . . .	27
3.4.1	Klasifikasi Data Objek . . . . .	27
3.4.2	Relasi Antar Objek . . . . .	30
<b>4</b>	<b>IMPLEMENTASI SISTEM DAN HASIL</b>	<b>31</b>
4.1	Implementasi Sistem . . . . .	31
4.1.1	Pengumpulan Data . . . . .	31
4.1.2	Pembacaan Data . . . . .	31
4.1.3	Pemberitahuan . . . . .	32
4.2	Hasil . . . . .	32
4.2.1	Hasil Antarmuka Aplikasi . . . . .	32
4.2.2	Hasil Pemberitahuan . . . . .	37
<b>5</b>	<b>PENGUJIAN DAN ANALISA</b>	<b>39</b>
5.1	Metode Pengujian . . . . .	39
5.2	<i>User Acceptance Test</i> (UAT) . . . . .	40
5.3	Waktu Eksekusi . . . . .	44
5.3.1	Waktu Eksekusi Unggah dan Unduh Data . .	44
5.3.2	Waktu Eksekusi Secara Bersamaan . . . . .	47
<b>6</b>	<b>PENUTUP</b>	<b>51</b>
6.1	Kesimpulan . . . . .	51
6.2	Penelitian Selanjutnya . . . . .	51
	<b>DAFTAR PUSTAKA</b>	<b>53</b>
	<b>Biografi Penulis</b>	<b>55</b>

# DAFTAR GAMBAR

2.1	Daftar cara autentifikasi menggunakan Firebase Au- thentication . . . . .	7
2.2	Contoh struktur JSON Tree . . . . .	10
2.3	Contoh denormalisasi data pada JSON Tree . . . . .	11
2.4	Contoh denormalisasi data pada JSON Tree . . . . .	12
3.1	Ilustrasi desain sistem yang dikembangkan . . . . .	16
3.2	<i>Layer</i> pengembangan sistem . . . . .	16
3.3	Alur proses persiapan data . . . . .	17
3.4	Alur proses pembacaan data . . . . .	18
3.5	Alur proses pengiriman pemberitahuan . . . . .	19
3.6	<i>Use Case diagram</i> . . . . .	19
3.7	<i>Activity diagram</i> untuk proses <i>login</i> dan registrasi .	21
3.8	<i>Activity diagram</i> untuk aktivitas Unggah ternak ba- ru, melihat detail data ternak dan melihat daftar ternak	22
3.9	<i>Activity diagram</i> untuk aktivitas update data ternak, hapus data ternak dan melihat laporan ternak . . .	23
3.10	Tampilan (a) <i>Login</i> dan (b) <i>Register</i> . . . . .	24
3.11	Tampilan (a) daftar ternak, (b) daftar dokter hewan dan (c) profil pengguna . . . . .	24
3.12	Tampilan untuk (a) ternak baru dan (b) detail ternak	25
3.13	Tampilan Laporan Ternak . . . . .	26
3.14	Struktur objek (a) <i>Cattle</i> , (b) <i>User</i> dan (c) <i>veterinary</i>	27
3.15	Struktur objek <i>user-cattles</i> . . . . .	30
4.1	Tampilan (a) <i>Login</i> dan (b) <i>Register</i> . . . . .	33
4.2	Tampilan (a) <i>Fragment</i> Ternak, (b) <i>Fragment Vete- riner</i> dan (c) <i>Fragment</i> Profil . . . . .	34
4.3	Tampilan antarmuka Tambah Ternak . . . . .	35
4.4	Tampilan antarmuka Detail Ternak . . . . .	36
4.5	Tampilan antarmuka Perbaharuan Ternak . . . . .	36
4.6	Tampilan Laporan Ternak . . . . .	37
4.7	Tampilan pemberitahuan pembaharuan ternak . . .	37
5.1	Grafik hasil pengujian waktu eksekusi unggah data .	45

5.2	Grafik hasil pengujian waktu eksekusi <i>authentication</i> pengguna . . . . .	46
5.3	Grafik hasil pengujian waktu eksekusi pengunduhan data . . . . .	48
5.4	Grafik hasil pengujian waktu eksekusi <i>authentication</i> data secara bersamaan . . . . .	49
5.5	Grafik hasil pengujian waktu eksekusi pengunduhan data secara bersamaan . . . . .	50

## DAFTAR TABEL

3.1	Detail <i>use case diagram</i> . . . . .	20
3.2	Struktur Objek Cattle/init-data . . . . .	28
3.3	Struktur Objek Cattle/update-data-i . . . . .	29
3.4	Struktur Objek <i>User</i> . . . . .	29
3.5	Struktur Objek <i>Veterinary</i> . . . . .	29
5.1	Hasil <i>User Acceptance Test</i> (UAT) . . . . .	43
5.2	Hasil pengujian waktu eksekusi unggah data . . . . .	45
5.3	Tabel hasil pengujian waktu eksekusi <i>authentication</i> data . . . . .	46
5.4	Tabel hasil pengujian waktu eksekusi unduh data . . . . .	47
5.5	Hasil pengujian waktu eksekusi <i>authentication</i> data secara bersamaan . . . . .	49
5.6	Hasil pengujian waktu eksekusi unduh data secara bersamaan . . . . .	50

*Halaman ini sengaja dikosongkan*



## DAFTAR KODE

2.1	Contoh <i>rule</i> akses tanpa mewajibkan autentifikasi user	8
2.2	Menambahkan <i>repository</i> Graph View pada andorid	13
2.3	Menambahkan <i>repository</i> Picasso pada andorid . . .	14

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

### 1.1 Latar belakang

Perkembangan teknologi informasi yang sangat pesat belakangan memicu munculnya tren baru dalam dunia bisnis. Banyak bisnis yang berubah dengan menambahkan aspek teknologi informasi didalamnya. Jika sebelumnya untuk memesan taksi harus melakukan panggilan telpon, kini hanya dengan menggunakan satu kali sentuhan jari pada telepon pintar atau *smartphone*. Sayangnya, beberapa sektor bisnis masih belum mengikuti perkembangan tersebut. Salah satunya adalah sektor peternakan. Menurut Direktorat Jenderal Peternakan Indonesia setiap tahun membutuhkan 653,98 ton daging sapi dengan kapasitas produksi nasional sebesar 418,547 ton atau sekitar 64% dari kebutuhan daging sapi nasional sedangkan sisanya sebesar 243,43 ton ditutupi dengan melakukan impor.<sup>[1]</sup> Disisi lain Indonesia memiliki potensi dalam bidang peternakan yang sangat besar dan masih dapat ditingkatkan. Salah satu masalah yang dialami oleh peternakan di Indonesia adalah pemantauan hewan ternak. Dengan adanya sistem pemantauan hewan ternak yang baik, peternakan dapat melakukan analisa terhadap perkembangan ternak dan dapat melakukan evaluasi terhadap cara memperlakukan ternak.

Kondisi yang ada saat ini di peternakan masih melakukan pencatatan secara manual. Sehingga data yang dikumpulkan rawan tercecer dan hilang. Selain itu pada beberapa peternakan dengan model intiplasma (dimana peternak tersebar di beberapa daerah dengan sebuah peternakan inti), pencatatan tersebut semakin sulit karena jauhnya jarak antar sub peternakan. Beberapa sistem pemantauan perkembangan hewan ternak telah dibuat sebelumnya, namun sistem yang ada masih menggunakan *desktop* sementara tidak semua peternak memiliki dan mampu mengoperasikannya. Oleh karena itu

dibutuhkan sebuah sistem untuk pemantauan ternak yang mampu menyimpan data secara terpusat, terintegrasi dengan jaringan, mudah digunakan, mudah untuk diakses kembali dan dapat digunakan secara multi-lokasi guna memudahkan pemantauan hewan ternak.

Dalam tugas akhir ini akan dikembangkan sebuah aplikasi bergerak untuk mempermudah pengawasan peternakan berbasis komputasi awan. Dengan menggunakan komputasi awan data akan tersimpan pada *server* yang terpusat dapat diakses kembali oleh *client* melalui jaringan. Hal ini mempermudah melakukan pengembangan sistem secara terintegrasi kepada seluruh *client*. Dengan bantuan jaringan internet pengawasan peternakan dapat dilakukan secara *remote* dan *realtime* atau semi *realtime*.

## 1.2 Permasalahan

Pemantauan perkembangan ternak harus dilakukan dengan sistem yang terintegrasi dan mudah dioperasikan oleh peternak. Pencatatan perkembangan ternak yang dilakukan secara manual mengakibatkan rawan terjadinya kehilangan data. Disamping itu sistem yang ada haruslah sistem yang *user friendly* bagi peternak. Aplikasi pemantauan ternak yang telah dikembangkan berbasis *desktop* dimana tidak semua peternak mampu mengoperasikannya. Karena dalam berbasis *desktop* aplikasi semacam ini tidak efisien untuk digunakan dilapangan dimana data harus disimpan secara langsung agar tidak mengulang pekerjaan. Selain itu data yang tidak tersimpan secara terpusat akan menyulitkan peternak dalam melakukan analisa data.

## 1.3 Tujuan

Tujuan utama dari tugas akhir ini adalah untuk mengembangkan *platform* pemantauan pertumbuhan ternak pada peternakan berbasis komputasi awan yang mampu,

1. Menyimpan data ternak dalam komputasi awan
2. Menampilkan data ternak mutakhir
3. Melakukan perbaharuan data ternak secara *realtime*
4. Memberikan pemberitahuan kepada pengguna tentang keadaan perkembangan ternak

5. Mempermudah hubungan antara dokter hewan dengan pengguna
6. Menyajikan laporan perkembangan ternak.

## 1.4 Batasan masalah

Untuk memfokuskan permasalahan yang akan diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut diantaranya adalah:

1. *Cloud Platform* yang dikembangkan menggunakan *Google Firebase API*
2. Aplikasi *front-end* dikembangkan berbasis Android
3. Data yang digunakan sebagai masukan berasal dari catatan peternak secara manual dengan mengambil sample
4. Jenis hewan yang dijadikan objek penelitian adalah Sapi berjenis *Limosin*.

## 1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan  
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Tinjauan Pustaka  
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait fitur *Google Firebase API*, JSON, komputasi awan dan teori-teori penunjang lainnya.
3. BAB III Perancangan Sistem dan Implementasi  
Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang dibuat dapat ter-

lihat dan mudah dibaca untuk implentasi pada pelaksanaan penelitian tugas akhir.

4. BAB IV Implementasi Sistem dan Hasil

Bab ini berisi tentang implementasi sistem yang telah dikembangkan. Implementasi proses pengumpulan data, desain basis data dan desain antar muka pengguna akan dijelaskan pada bagian ini. Sehingga dapat dilihat hasil aplikasi yang telah dikembangkan baik secara proses maupun secara visual.

5. BAB V Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Hasil pengujian terhadap pengguna akan dijelaskan pada bab ini. Spesifikasi tipe jaringan seluler yang digunakan dan lokasi tempat pengujian juga dijelaskan pada bab ini. Bebandingan kecepatan eksekusi pada berbagai perintah akan dibandingkan pada bab ini sehingga dapat diketahui jaringan yang optimal untuk sistem yang dikembangkan.

6. BAB VI Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

## BAB 2

### TINJAUAN PUSTAKA

Demi mendukung pengerjaan penelitian ini, dibutuhkan teori yang menunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

#### 2.1 Komputasi Awan

Komputasi awan atau *cloud computing* menurut definisi dari NIST (National Institute of Standard Technology) merupakan model yang memungkinkan penggunaan dari berbagai tempat secara bersamaan atau *ubiquitous*, penggunaan secara mudah, penggunaan jaringan sesuai kebutuhan pada kumpulan sumber daya komputasi yang dapat dikonfigurasi (misalnya jaringan, *server*, penyimpanan data, aplikasi dan layanan) sehingga dapat tersedia dan diterbitkan dengan usaha sekecil mungkin tanpa harus melakukan interaksi dengan penyedia layanan.<sup>[2]</sup>

Komputasi awan dapat dibagi menjadi dua bagian: *front end* dan *back end*.<sup>[3]</sup> Keduanya terhubung satu sama lain melalui jaringan, biasanya berupa jaringan internet. *Front end* adalah apa yang disajikan kepada pengguna, sedangkan *back end* adalah sistem yang berada di awan (*cloud system*). *Front end* haruslah berada pada komputer pengguna yakni aplikasi yang dibutuhkan untuk melakukan akses pada *cloud* sedangkan *back end* merupakan layanan komputasi awan itu sendiri yang terdiri dari berbagai macam komputer, *server* dan penyimpanan data.<sup>[4]</sup>

Pemantauan lalu lintas, pengelolaan sistem dan permintaan klien dilakukan oleh *server* utama. Hal ini dilakukan berdasarkan protokol tertentu dan menggunakan perangkat lunak khusus yang disebut *middleware*. *Middleware* memungkinkan komputer pada jaringan untuk berkomunikasi satu sama lain.<sup>[3][4]</sup> Dalam tugas akhir ini *middleware* yang digunakan adalah Google Firebase API yang akan dijelaskan selanjutnya.

## 2.1.1 Manfaat Komputasi Awan

### a Pengelolaan yang Mudah

Perawatan infrastruktur baik hardware maupun software menjadi lebih mudah.<sup>[4]</sup> Infrastruktur pada umumnya telah dibangun oleh penyedia *cloud services* sehingga tidak perlu membangun infrastruktur sendiri. Oleh karena itu perawatan yang rumit tidak diperlukan. Sedangkan pada sisi klien yang diperlukan adalah aplikasi atau *web browser* dengan koneksi internet.

### b Mengurangi Biaya

Komputasi awan mampu mengurangi biaya yang dibutuhkan dalam bidang IT. Hal ini karena tidak diperlukannya pembangunan infrastruktur secara mandiri. Sehingga tenaga yang dibutuhkan untuk pembangunan sistem ini pun tidak diperlukan. Sebagai contoh adalah layanan email yang dapat dilakukan secara gratis melalui layanan GMAIL, YAHOO, dsb.

### c Penanggulangan Bencana

Bencana terjadi secara tidak terduga, bagi sebagian orang kehilangan handphone atau laptop bisa menjadi bencana karena data penting berada pada perangkat-perangkat tersebut. Namun dengan menggunakan komputasi awan bencana kehilangan data dapat dihindari, beberapa penyedia layanan telah menyediakan *backup* yang dapat membantu mengembalikan data.

### d *Green Computing*

Apabila setiap organisasi membangun infrastruktur mereka sendiri tentu akan menghasilkan dampak negatif kepada lingkungan. Salah satunya adalah konsumsi energi listrik yang terbuang untuk menjaga agar infrastruktur tetap bekerja. Dengan penggunaan komputasi awan hal semacam ini dapat dikurangi. Sehingga mendorong penghematan energi dan pelestarian lingkungan.<sup>[4]</sup>










## 2.2 Google Firebase

Firebase dibangun oleh Andrew Lee dan James Tamplin pada tahun 2011 yang kemudian berpindah tangan kepada Google pada Oktober 2014. Firebase merupakan salah satu BaaS (*Backend as a Service*) yang memudahkan pengembangan aplikasi multiplatform baik iOS, Android maupun web apps.<sup>[5][6]</sup> Firebase menyediakan layanan yang mampu melakukan sinkronisasi data otomatis, autentifikasi, penyimpanan data *real-time* hingga *cloud function* yang memungkinkan pengembang untuk membangun program komputasi awan tanpa *server*.

### A Firebase Authentication

Firebase authentication adalah layanan *backend* dengan SDK yang sederhana yang memungkinkan pengembang aplikasi untuk membangun sistem autentifikasi instan dan dilengkapi dengan desain UI. Dalam Firebase authentication terdapat mekanisme *username* dan *password*, selain itu terdapat pula terintegrasi *social login* (Facebook dan Google Plus) serta integrasi dengan pihak ketiga lain seperti pada gambar 2.1.<sup>[7]</sup>

Penyedia proses masuk	
Penyedia	Status
 Email/Sandi	Diaktifkan
 Ponsel	Dinonaktifkan
 Google	Dinonaktifkan
 Facebook	Dinonaktifkan
 Twitter	Dinonaktifkan
 GitHub	Dinonaktifkan
 Anonim	Dinonaktifkan

**Gambar 2.1:** Daftar cara autentifikasi menggunakan Firebase Authentication

### B Firebase Realtime Database

Kebutuhan akan *realtime database* semakin meningkat, aplikasi perpesanan contohnya sangat memerlukan penyimpanan

data yang dapat diakses secara *realtime* karena pengguna pada umumnya menginginkan informasi yang cepat. Firebase Real-time Database memberikan layanan untuk menyimpan dan menyinkronkan data dengan menggunakan NoSQL *cloud database*. Data yang telah disimpan akan tersinkronasi pada setiap klien secara langsung dan tetap tersedia ketika aplikasi klien sedang *offline*. Realtime Database juga memiliki kemampuan untuk menyinkronkan perubahan data lokal dengan pembaruan jarak jauh yang terjadi saat klien *offline*, dengan menggabungkan perbedaan apa pun secara otomatis.<sup>[8]</sup>

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, bernama *Firebase Realtime Database Security Rules*, untuk mendefinisikan bagaimana struktur data Anda harus disusun serta kapan data bisa dibaca atau ditulis. Ketika terintegrasi dengan Firebase Authentication, pengembangan bisa menentukan siapa yang memiliki akses ke data, dan bagaimana mereka dapat mengaksesnya.<sup>[8]</sup> Contoh aturan ini seperti pada kode 2.1 yang menunjukkan bebas akses penulisan dan pembacaan data kepada semua pengguna tanpa autentifikasi.

```
{
  "rules": {
    "foo": {
      ".read": true,
      ".write": true
    }
  }
}
```

**Kode 2.1:** Contoh *rule* akses tanpa mewajibkan autentifikasi user

Data yang disimpan pada Firebase Realtime database berbasis data NoSQL dalam bentuk objek JSON. Firebase mengklaim bahwa sistem ini memiliki optimalisasi dan fungsionalitas yang berbeda dibandingkan dengan basis data relasional. Realtime Database API didesain untuk hanya memperbolehkan operasi yang bisa dieksekusi dengan cepat. Firebase juga mengklaim bahwa sistem ini dapat melayani jutaan pengguna tanpa

mengorbankan daya respons.<sup>[8]</sup>

### C **Firestore Storage**

Firestore Storage adalah layanan untuk pengembang aplikasi yang perlu menyimpan dan menyediakan konten buatan pengguna, seperti foto atau video. Firestore Storage menyimpan file di bucket *Google Cloud Storage*, yang berarti bahwa dapat diakses melalui Firestore dan Google Cloud API. Hal ini memberikan fleksibilitas bagi klien untuk mengunggah dan mengunduh file dari aplikasi klien melalui Firestore dan melakukan pemrosesan sisi *server* seperti filter gambar atau transkode video menggunakan *Google Cloud Platform* maupun *Firestore Cloud Function*.<sup>[9]</sup>

Firestore Storage juga telah terintegrasi dengan Firestore Authentication untuk mengidentifikasi pengguna. Sehingga pengembang dapat mengatur akses pada satu file atau kumpulan file yang telah disimpan menjadi bersifat publik maupun pribadi.<sup>[9]</sup>

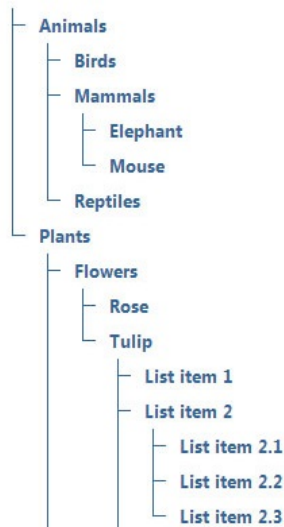
**D Firestore Cloud Function** Firestore Cloud Function memungkinkan pengembang untuk membangun program awan tanpa memiliki server atau *serverless*. Cloud Function menggunakan Node.js sebagai lingkungan pengembangan. Cloud Function akan menjalankan program pada *backend* secara otomatis ketika dipicu oleh fitur firestore (seperti *authentication*, *realtime database*, *storage* dan *cloud messaging*) maupun melalui *HTTP Request*.<sup>[10]</sup> *Life cycle* pengembangan sebuah fungsi pada Cloud Function adalah sebagai berikut,

- (a) Pengembang menulis kode untuk fungsi baru, memilih *event provider* (seperti Database Realtime), dan menentukan kondisi di mana fungsi harus dijalankan
- (b) Pengembang *me-deploy* fungsinya, dan Firestore menghubungkannya ke *event provider* yang dipilih
- (c) Saat *event provider* menghasilkan acara yang sesuai dengan kondisi fungsi, kode akan dijalankan

- (d) Saat pengembang memperbarui fungsinya dengan menerapkan kode yang diperbarui, semua kode untuk versi lama dibersihkan dan diganti dengan kode baru
- (e) Saat pengembang menghapus fungsinya, semua kasus dibersihkan dan koneksi antara fungsi dan penyedia acara akan dihapus.

## 2.3 Membuat Struktur Basisdata pada JSON Tree

Data yang disimpan dalam *cloud* adalah dalam bentuk objek JSON atau dalam struktur *JSON Tree*. Berbeda dengan struktur tabel relasional yang memiliki tabel dan rekaman, pada JSON data yang ditambahkan ke JSON tree, akan menjadi simpul dalam struktur JSON yang telah ada seperti terlihat pada gambar 2.2.<sup>[8]</sup> Dengan demikian relasi antar data merupakan simpul itu sendiri.



**Gambar 2.2:** Contoh struktur JSON Tree

1. Menghindari Data Bersarang *Nested Data*  
 Dalam Firebase pengguna memungkinkan untuk memiliki 32 tingkat simpul, meskipun demikian membuat struktur dengan terlalu banyak simpul akan menimbulkan masalah. Pasalnya

ketika pengguna mengakses sebuah simpul, pengguna juga dapat mengambil semua data yang terdapat pada simpul anaknya. Begitu pula ketika pengembang memberikan akses data pada suatu simpul artinya juga memberikan akses pada simpul anaknya. Oleh karena itu sebaiknya menghindari data yang bersarang.<sup>[8]</sup>

## 2. Meratakan Struktur Data

Menyimpan data dalam *path* terpisah atau dapat disebut denormalisasi data adalah hal yang normal dilakukan pada struktur data JSON.<sup>[11]</sup> Dengan demikian data tersebut akan lebih mudah diunduh dengan panggilan terpisah. Sebagai contoh adalah gambar 2.3 didalamnya terdapat data yang diduplikasi. Hal ini dilakukan untuk mempermudah pengunduhan data dan juga untuk mempermudah skalasi data berikutnya.

```
{
  users: {
    user1: {
      name: "Alice"
    },
    user2: {
      name: "Bob"
    }
  },
  links: {
    link1: {
      title: "Example",
      href: "http://example.org",
      submitted: "user1"
    }
  },
  comments: {
    comment1: {
      link: "link1",
      body: "This is awesome!",
      author: "user2"
    }
  }
}
```

**Gambar 2.3:** Contoh denormalisasi data pada JSON Tree

Menduplikasi data memang bukan hal yang baik dilakukan, namun demi membangun basisdata yang dapat diskalaskan

denormalisasi data adalah cara yang terbaik.

### 3. Membuat Data yang Dapat Diskalasikan

Relasi antar objek pasti selalu ada dalam membangun basidata. Akan lebih mudah apabila relasi ini ditempatkan pada simpul anak dari sebuah objek. Namun hal ini akan menjadi rumit apabila relasi tersebut bersifat dinamis atau berubah-ubah. Sebagai contoh pada gambar 2.4 terlihat bahwa adanya redundansi data pada simpul `users/alovelance` yang juga menyimpang daftar grup sementara pada simpul grup juga terdapat daftar anggota. Namun redundansi ini dibutuhkan dalam hal ini karena hubungan antara keduanya dinamis, *users* dapat menghapus dan menambahkan group, begitu pula anggota grup dapat berubah-ubah.

```
{
  users: {
    user1: {
      name: "Alice"
    },
    user2: {
      name: "Bob"
    }
  },
  links: {
    link1: {
      title: "Example",
      href: "http://example.org",
      submitted: "user1"
    }
  },
  comments: {
    comment1: {
      link: "link1",
      body: "This is awesome!",
      author: "user2"
    }
  }
}
```

**Gambar 2.4:** Contoh denormalisasi data pada JSON Tree

## 2.4 Library Android Pendukung

Dalam pengembangan aplikasi *frontend* IDE (*Integrated Development Environment*) yang digunakan adalah Android Studio. Untuk mendukung pengembangan aplikasi ini, beberapa komponen tidak dimiliki oleh Android Studio. Salah satu contohnya adalah untuk menampilkan URL menjadi gambar pada UI. Contoh lainnya adalah untuk membuat grafik, bagan atau skema pada UI juga tidak secara langsung didukung oleh Android Studio. Oleh karena itu dibutuhkan beberapa *library* tambahan untuk mempermudah pekerjaan tersebut.

### 2.4.1 Android Graph View

Android Graph View adalah salah satu *library* untuk membuat diagram, grafik maupun bagan dengan menggunakan program (*programmatically*). Graph View memudahkan pengembang untuk membuat grafik dengan pola yang dapat disesuaikan dengan keinginan pengembang. Graph View juga memungkinkan pengembang untuk membuat grafik secara realtime yang dapat dikombinasikan dengan realtime database. Selain itu pada Graph View memungkinkan untuk menggabungkan beberapa informasi data dalam satu buah grafik sehingga memudahkan pengembang apabila ingin menampilkan beberapa informasi sekaligus sebagai perbandingan dalam sebuah grafik. Sama seperti *library* android pada umumnya untuk menggunakan Graph View cukup dengan menambahkan *repository*-nya pada *dependency* android seperti kode 2.2 ini atau dengan menambahkan *file* .jar pada folder *libs*.

```
compile 'com.jjoe64:graphview:4.2.1'
```

**Kode 2.2:** Menambahkan *repository* Graph View pada android

### 2.4.2 Picasso

Picasso merupakan *library* yang dikembangkan dibawah lisensi *Apache* yang berguna untuk menampilkan URL kedalam bentuk gambar. Picasso memungkinkan loading gambar dengan menggu-

nakan satu baris kode yang singkat. Picasso juga mengatasi 'jebakan' umum ketika memuat gambar pada Android, dimana dalam beberapa kasus gambar terunduh secara otomatis. Picasso melakukan *handling* otomatis untuk menggagalkan pengunduhan. Picasso juga secara otomatis mengatur memori dan *disk caching* untuk mengunduh gambar. Selain itu Picasso juga memberikan fitur untuk mengubah gambar agar sesuai dengan *layout* yang telah disediakan pada UI. Seperti Graph View untuk menggunakan Picasso pengembang harus menambahkan *repository*-nya pada *dependency* android seperti kode 2.3 ini atau dengan menambahkan *file* .jar pada folder *libs*.

```
compile 'com.squareup.picasso:picasso:2.5.2'
```

**Kode 2.3:** Menambahkan *repository* Picasso pada android



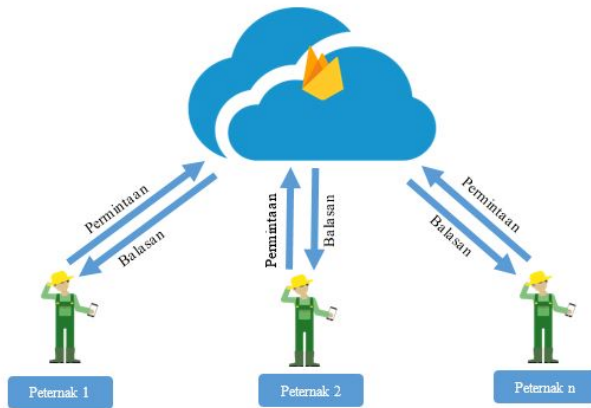
## BAB 3

# DESAIN SISTEM

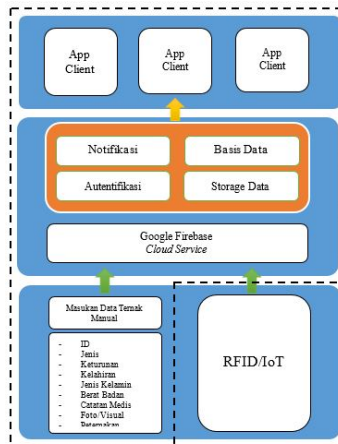
Penelitian ini dilaksanakan dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dan rancangan sistem yang kemudian diimplementasikan dalam penelitian ini. Dalam hal ini adalah perancangan aplikasi berbasis desain UML, perancangan antar muka pengguna, perancangan algoritma dan perancangan basis data.

### 3.1 Desain Sistem

Tujuan dari penelitian ini adalah untuk meningkatkan efisiensi para peternak dalam melakukan pemantauan pertumbuhan hewan ternak mereka melalui aplikasi bergerak berbasis komputasi awan. Bentuk sistem sendiri merupakan sebuah aplikasi yang dapat diakses oleh banyak pengguna dari berbagai tempat secara bersamaan seperti ditunjukkan pada gambar 3.1. Aplikasi klien akan dibangun berbasis Android dan dikembangkan dalam IDE Android Studio. Kemudian untuk *backend* sistem sebagai komputasi awan menggunakan *platform* yang dikembangkan oleh Google yakni Google Firebase dengan memanfaatkan API yang sudah ada. Sistem yang akan dikembangkan ditunjukkan pada gambar 3.2. Terdapat tiga layer yakni pengumpulan data, *server* (Firebase) dan klien.



**Gambar 3.1:** Ilustrasi desain sistem yang dikembangkan



**Gambar 3.2:** Layer pengembangan sistem

### 3.1.1 Proses Pengumpulan Data

Pada penelitian kali ini pengumpulan data dilakukan secara manual. Data yang akan dijadikan masukan untuk sistem berupa ID, jenis, umur, warna kulit, berat badan, tinggi badan, lingkaran badan dan foto dari sapi. Data diambil dari pengamatan terha-

dap hewak ternak secara langsung. Setelah diperoleh data yang dibutuhkan, data fisik ini kemudian diubah menjadi data digital dengan menggunakan aplikasi bergerak. Kemudian barulah data siap untuk disimpan dalam basis data.. Dalam proses ini data yang diambil tidak hanya satu kali melainkan diambil secara berkala yakni untuk melihat perkembangan hewan ternak. Data yang pertama kali diambil disebut sebagai data inisialisasi sedangkan data perbaruaran disebut sebagai data perbaruan. Pada data perbaruan parameter-parameter yang tidak berubah seperti ID dan jenis tidak akan dilakukan pemngumpulan ulang. Alur proses persiapan data ini ditunjukkan pada gambar 3.3.



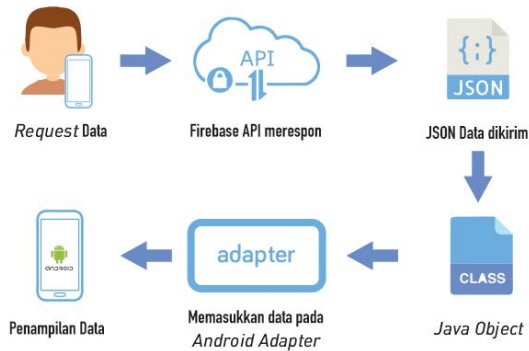
**Gambar 3.3:** Alur proses persiapan data

### 3.1.2 Proses Pembacaan Data

Data yang telah disimpan pada basis data dapat dipanggil kembali ketika pengguna ingin melihat data perkembangan ternak terkini maupun ketika pengguna ingin melihat laporan secara keseluruhan. Proses pemanggilan ini menggunakan API yang disediakan oleh Google Firebase. API yang dipanggil akan memberikan respon dengan mengirimkan data yang diinginkan dalam format *Java Object*. Untuk menampilkan objek pada aplikasi pengguna, telah disediakan *class* dan *adapter* sebagai tempat penyimpanan sementara. Hal ini dilakukan karena data yang dipanggil hanyalah sebatas proses pembacaan saja sehingga tidak disimpan pada perangkat. Proses pembacaan ini dapat dilihat pada gambar 3.4.

### 3.1.3 Proses Pemberitahuan

Pada saat pengguna melakukan kegiatan pada basis data, sistem akan mengirimkan tanggapan kepada pengguna berupa pemberitahuan. Pemberitahuan ini dikembangkan dengan menggunakan *Firebase Cloud Function* dengan memanfaatkan perubahan pada



**Gambar 3.4:** Alur proses pembacaan data

*Firebase real time database* sebagai pemicu. Pemberitahuan yang dikembangkan terbatas pada dua kegiatan yakni ketika pengguna menambahkan ternak baru dan ketika pengguna menambahkan pembaruan data ternak. Kedua kegiatan diatas akan menambahkan data baru pada lokasi tertentu pada basis data, ketika ini terjadi *cloud function* akan mendeteksi perubahan ini dan memberikan mengirimkan *request* kepada *Firebase Cloud Messaging* untuk mengirimkan pemberitahuan kepada pengguna bahwa transaksi yang dilakukan telah berhasil. Proses pemberitahuan ini disajikan pada gambar 3.5.

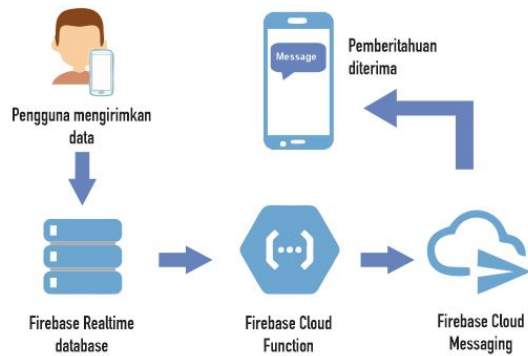
Proses pembacaan ini terjadi pula ketika pengguna ingin melihat laporan perkembangan secara keseluruhan. Data yang telah dikumpulkan secara berkala disusun berdasarkan waktu dan dikelompokkan sesuai jenis parameternya. Kemudian ditampilkan pada aplikasi pengguna.

## 3.2 Desain Aplikasi Berbasis UML

Dalam sub bab ini dijelaskan tentang desain sistem berdasarkan konsep berorientasi objek dalam beberapa diagram sebagai berikut,

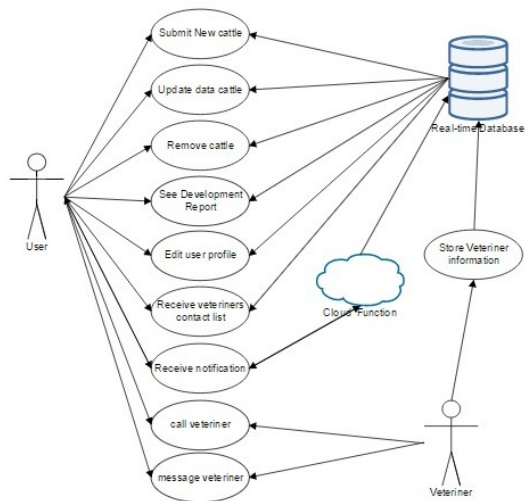
### 1. Use Case Diagram

Hubungan antara aktor dengan *use case* pada sistem yang dibuat disajikan pada gambar 3.6 sedangkan penjelasannya



**Gambar 3.5:** Alur proses pengiriman pemberitahuan

disajikan pada tabel 3.1.



**Gambar 3.6:** Use Case diagram

## 2. Activity Diagram

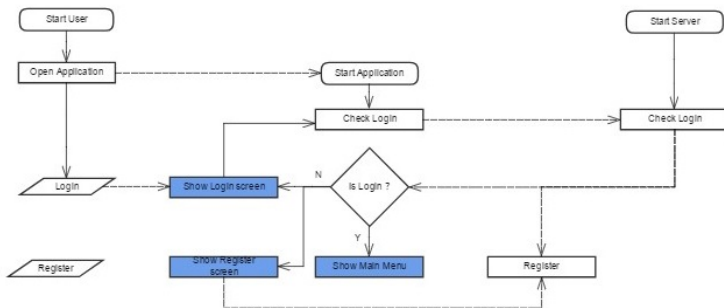
*Activity Diagram* menunjukkan bagaimana interaksi antara pengguna terhadap sistem baik *front-end* maupun *back-end*.

**Tabel 3.1:** Detail *use case diagram*

<i>Use Case</i>	<b>Aktor</b>	<b>Keterangan</b>
Masuk	peternak, basidata	Interaksi antara peternak dan basidata ketika melakukan proses <i>login</i>
Registrasi	peternak, basidata	Interaksi antara peternak dan basidata ketika melakukan proses registrasi
Unggah ternak baru	peternak, basidata	Interaksi ketika peternak melakukan proses pengunggahan data ternak baru
Perbarui data ternak	peternak, basidata	Interaksi peternak dan basidata saat proses perbaruan data ternak
Hapus data ternak	peternak, basidata	Interaksi peternak menghapus data ternak dari basidata
Detail ternak	peternak, basidata	Interaksi ketika peternak membaca informasi detail ternak dari basidata
Laporan perkembangan	peternak, basidata	Interaksi ketika peternak membaca laporan ternak dari basidata
Mengubah profil pengguna	peternak, basidata	Interaksi ketika peternak memperbarui data pada profilnya
Panggilan dokter hewan	peternak, basidata, dokter hewan	Interaksi ketika peternak ingin melakukan panggilan kepada dokter hewan
SMS dokter hewan	peternak, basidata, dokter hewan	Interaksi ketika peternak ingin mengirimkan SMS kepada dokter hewan
Notifikasi	peternak, basidata, cloud function	Notifikasi dikirim secara otomatis dipicu oleh peternak yang melakukan perubahan pada basidata

Diagram ini akan memperlihatkan tentang alur aktivitas pengguna mulai dari pertama kali menggunakan aplikasi hingga keluar dari aplikasi serta respon yang diberikan oleh setiap bagian sistem kepada pengguna.

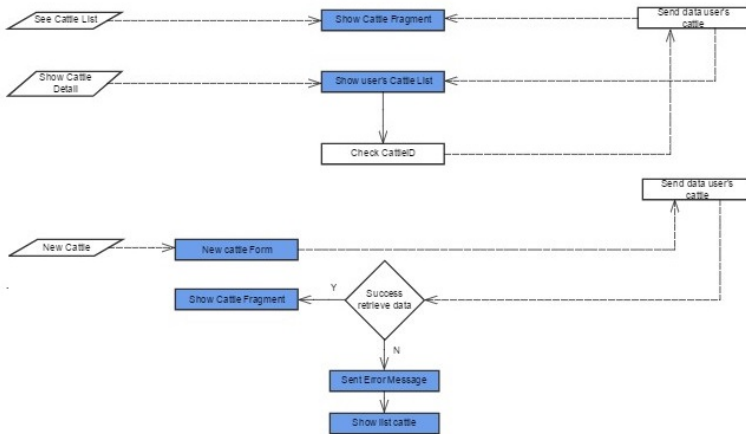
- (a) *Activity Diagram* saat pengguna pertama kali menggunakan aplikasi ditunjukkan pada gambar 3.7. Aktivitas yang dilakukan pengguna pertama kali menggunakan aplikasi termasuk diantaranya adalah aktivitas *login* dan registrasi.
- (b) *Activity Diagram* pada gambar 3.8 menunjukkan aktivitas yang pengguna dapat lakukan ketika berada pada menu utama setelah terautentikasi. Aktivitas yang dapat dilakukan pada tampilan ini adalah melihat daftar ternak, mengunggah ternak baru dan melihat detail data ternak.
- (c) *Activity Diagram* pada gambar 3.9 menunjukkan alur ketika pengguna ingin menambahkan perbaruan data ternak, menghapus data ternak dan melihat laporan data ternak.



**Gambar 3.7:** *Activity diagram* untuk proses *login* dan registrasi

### 3.3 Desain Antarmuka Aplikasi

Pada bagian ini dijelaskan tentang desain antarmuka aplikasi yang dikembangkan. Sedikitnya ada sembilan desain antarmuka



**Gambar 3.8:** Activity diagram untuk aktivitas Unggah ternak baru, melihat detail data ternak dan melihat daftar ternak

untuk aplikasi ini sebagai berikut,

### 1. Tampilan *Login*

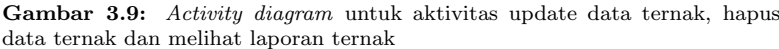
Tampilan *Login* merupakan tampilan yang digunakan untuk melakukan autentifikasi pengguna. Pada saat pertama kali penggunaan tampilan ini adalah tampilan yang pertama kali muncul. Namun setelah pengguna melakukan registrasi dan selama tidak melakukan *logout* pengguna akan langsung dibawa ke tampilan menu utama. Rancangan antarmuka ini ditunjukkan pada gambar 3.10a.

### 2. Tampilan *Register*

Tampilan *Register* merupakan tampilan yang digunakan untuk melakukan registrasi pengguna baru. Sehingga pengguna dapat menggunakan fitur pada aplikasi ini. Pada tampilan ini pengguna diminta untuk memasukkan beberapa data untuk registrasi seperti nama, email dan kata sandi. Rancangan antarmuka ini ditunjukkan pada gambar 3.10b.

### 3. Tampilan Menu Utama-Ternak

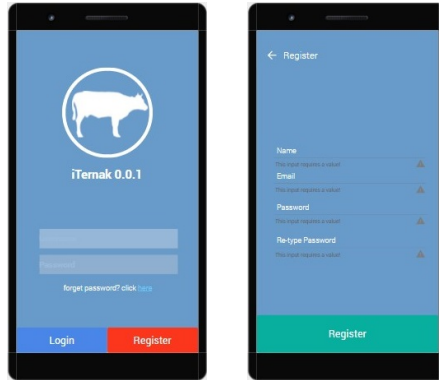




Ketika pengguna telah sukses melakukan registrasi pengguna akan dibawa ke tampilan ini. Tampilan ini menunjukkan daftar ternak yang dimiliki oleh pengguna. Dalam tampilan ini pengguna dapat melakukan penambahan ternak baru melalui *float action button* yang disediakan. Pengguna juga dapat melihat detail ternak yang dengan memilih salah satu daftar. Rancangan antarmuka ini ditunjukkan pada gambar 3.11a.

#### 4. Tampilan Menu Utama-Dokter Hewan

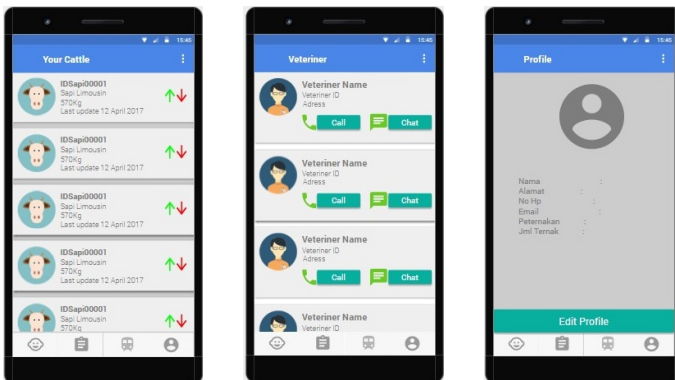
Dengan melakukan *swipe* pada menu utama pengguna dapat berpindah dari Menu Ternak ke Menu Dokter Hewan. Pada menu ini disajikan daftar dokter hewan yang dapat dihubungi oleh pengguna dalam keadaan tertentu. Pengguna dapat melakukan panggilan langsung maupun mengirimkan pesan



(a)

(b)

**Gambar 3.10:** Tampilan (a) *Login* dan (b) *Register*

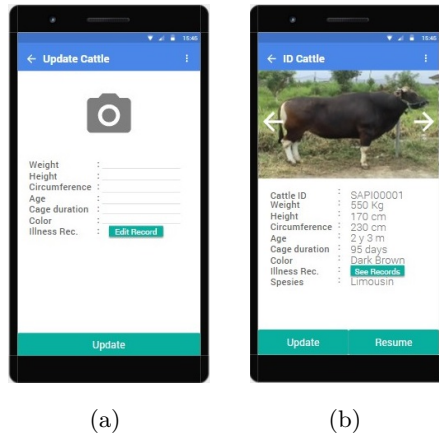


(a)

(b)

(c)

**Gambar 3.11:** Tampilan (a) daftar ternak, (b) daftar dokter hewan dan (c) profil pengguna



**Gambar 3.12:** Tampilan untuk (a) ternak baru dan (b) detail ternak

singkat kepada dokter hewan yang ada melalui menu ini. Rancangan antarmuka ini ditunjukkan pada gambar 3.11b.

#### 5. Tampilan Menu Utama-Profil

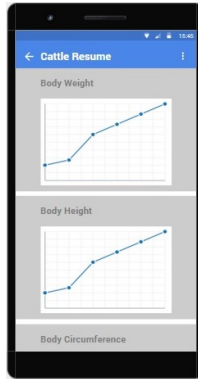
Bagian Menu utama yang ketiga adalah Menu Profil. Tampilan ini menunjukkan data profil pengguna seperti foto, nama, nomer hp, email dan jumlah ternak yang dimiliki. Pengguna juga dapat melakukan perubahan profil melalui menu ini. Rancangan antarmuka ini ditunjukkan pada gambar 3.11c.

#### 6. Tampilan Ternak Baru

Tampilan Ternak Baru akan muncul dengan dipicu oleh *floating action button* yang ada pada Tampilan Menu Utama. Tampilan ini berisi formulir untuk menambahkan informasi ternak baru pada basis data pengguna. Rancangan antarmuka ini ditunjukkan pada gambar 3.12a.

#### 7. Tampilan Perbarui Ternak

Tampilan Perbarui Ternak memiliki desain yang mirip dengan tampilan Ternak Baru. Tampilan ini berisi formulir yang untuk menambahkan perbaharuan informasi ternak tertentu.



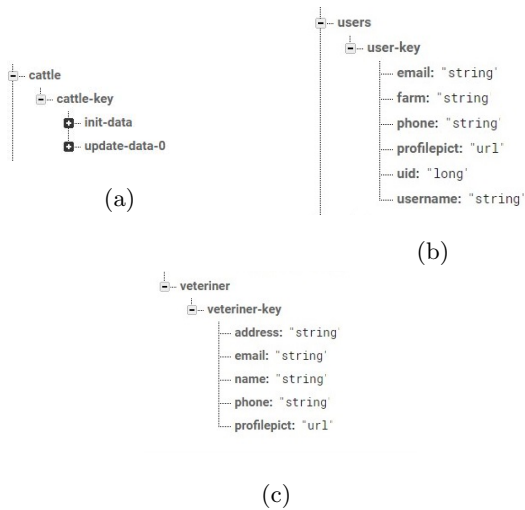
**Gambar 3.13:** Tampilan Laporan Ternak

#### 8. Tampilan Detail Ternak

Tampilan Detail Ternak adalah tampilan untuk menunjukkan detail informasi tentang ternak tertentu. Tampilan ini akan muncul ketika pengguna memilih salah satu daftar ternak pada Menu Utama-Ternak. Informasi yang disajikan pada tampilan ini merupakan informasi mutakhir yang diunggah oleh pengguna. Pada tampilan ini terdapat dua tombol navigasi untuk berpindah ke tampilan perbarui ternak dan tampilan laporan ternak. Rancangan antarmuka ini ditunjukkan pada gambar 3.12b.

#### 9. Tampilan Laporan Ternak

Tampilan Laporan Ternak berisi seluruh rangkuman informasi tentang ternak tertentu. Informasi ternak secara umum seperti usia, durasi pengembangan, warna kulit dan jenis ternak. Selain itu informasi seperti berat badan, tinggi badan dan lingkaran badan disajikan dalam bentuk grafik agar mudah dipahami. Sebagai tambahan terdapat pula hasil kalkulasi rata-rata pertumbuhan, pertumbuhan maksimal dan minimal untuk mempermudah pengguna dalam melakukan analisa pertumbuhan ternak. Terdapat pula informasi ternak secara umum seperti usia, durasi pengembangan, warna kulit dan jenis ternak. Rancangan antarmuka ini ditunjukkan pada gam-



**Gambar 3.14:** Struktur objek (a) *Cattle*, (b) *User* dan (c) *veterinary*

bar 3.13.

### 3.4 Desain Basis Data

Pada bagian ini dijelaskan tentang desain basis data yang digunakan. Struktur basis data yang digunakan adalah basis data NoSQL dengan sturktur JSON Tree. Bentuk JSON Tree berbeda dengan struktur tabel dimana skema relasi dihubungkan melalui *foreign key*, pada JSON Tree untuk menunjukkan relasi antar objek dapat berupa simpul anak dan dapat berupa denormalisasi data.

#### 3.4.1 Klasifikasi Data Objek

Data yang digunakan dalam penelitian ini dibagi menjadi tiga objek utama yakni objek ternak atau *cattle*, objek pengguna atau *user*, objek dokter hewan atau *veterinary*.

##### 1. Objek *Cattle*

Data ternak berisi informasi yang dibutuhkan untuk pengembangan ternak. Struktur objek ini disajikan seperti pada gambar 3.14a. Data ini dibagi menjadi dua yakni data inialisasi

**Tabel 3.2:** Struktur Objek *Cattle/init-data*

Nama Field	Tipe Data	Keterangan
id	string	ID Hewan
uid	string	User ID pemilik hewan
age	long	Usia hewan
color	string	Warna kulit hewan
circumstance	long	Lingkar badan hewan
weight	long	Berat badan hewan
height	long	Tinggi badan hewan
species	string	Jenis hewan
timestamp	long	Tanggal unggahan data
day_count	long	Durasi hewan digemukkan
imageURL	string	URL lokasi foto hewan disimpan pada <i>storage</i>
update_count	long	Jumlah perbaruan data

dan data perbaruan. Dalam inisialisasi memuat informasi keseluruhan informasi ternak ketika pertama kali dimasukkan kedalam sistem.

Data ini meliputi ID, jenis sapi, usia, tanggal masuk, warna kulit, berat badan, tinggi badan dan lingkar badan serta foto. Sedangkan data perbaruan adalah informasi yang diamati selama penggemukan yakni warna kulit, berat badan, tinggi badan, lingkar badan dan foto. Detail informasi yang disimpan dapat dilihat pada tabel 3.2 dan 3.3.

## 2. Objek *User*

Data pengguna disimpan dalam objek ini. Informasi yang disimpan adalah nama, alamat email, nomer telepon, fotoURL, nama peternakan dan uid. Struktur objek ini disajikan seperti pada gambar 3.14b dan lebih detail dijelaskan pada tabel 3.4.

## 3. Objek *Veterinary*

Data dokter hewan yang disimpan meliputi nama, nomer telepon, alamat dan fotoURL. Struktur objek ini disajikan seperti pada gambar 3.14c dan lebih detail dijelaskan pada tabel 3.5.

**Tabel 3.3:** Struktur Objek `Cattle/update-data-i`

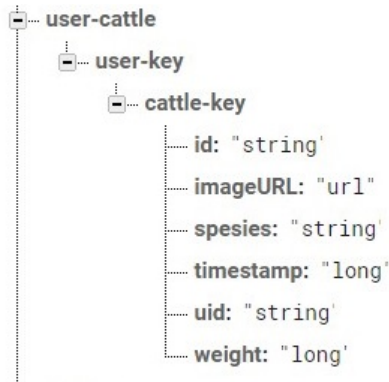
Nama Field	Tipe Data	Keterangan
id	string	ID Hewan
age	long	Usia hewan
color	string	Warna kulit hewan
circumstance	long	Lingkar badan hewan
weight	long	Berat badan hewan
height	long	Tinggi badan hewan
timestamp	long	Tanggal unggahan data
imageURL	string	URL foto hewan disimpan pada <i>storage</i>

**Tabel 3.4:** Struktur Objek `User`

Nama Field	Tipe Data	Keterangan
uid	string	ID pengguna
username	string	Nama pengguna
email	string	Alamat email pengguna
phone	string	Nomor telepon pengguna
farm	string	Nama peternakan pengguna
profilepict	string	URL foto profil pengguna disimpan pada <i>storage</i>

**Tabel 3.5:** Struktur Objek `Veterinary`

Nama Field	Tipe Data	Keterangan
username	string	Nama dokter hewan
email	string	Alamat email dokter hewan
phone	string	Nomor telepon dokter hewan
address	string	Alamat dokter hewan
profilepict	string	URL foto profil pengguna disimpan pada <i>storage</i>



**Gambar 3.15:** Struktur objek *user-cattles*

### 3.4.2 Relasi Antar Objek

Seperti telah disinggung diawal bab, untuk menunjukkan relasi antar objek pada struktur *JSON Tree* terkadang dibutuhkan denormalisasi data. Hal ini juga dilakukan pada penelitian kali ini. Relasi antara objek *Cattle* dengan *User* dibuat dengan membuat sebuah objek baru yakni *user-cattles*. Objek ini berisi informasi hewan ternak yang dimiliki oleh setiap pengguna. Objek ini juga bermanfaat untuk mempermudah *query* data untuk menampilkan daftar hewan ternak yang dimiliki oleh pengguna tanpa haru melakukan pencarian kedalam objek *Cattle*. Sebenarnya data terpenting untuk dimasukkan dalam objek ini hanyalah uid dan id *cattle* saja. Namun untuk kebutuhan lain seperti pada tampilan daftar ternak yang membutuhkan informasi *preview* pada *listview* sehingga beberapa informasi ditambahkan dalam objek ini. Struktur objek ini dapat dilihat pada gambar 3.15.



## **BAB 4**

### **IMPLEMENTASI SISTEM DAN HASIL**

#### **4.1 Implementasi Sistem**

Pada bagian ini dibahas tentang implementasi desain sistem berdasarkan proses kerjanya.

##### **4.1.1 Pengumpulan Data**

Proses pengumpulan data melalui beberapa tahapan sebelum dapat disajikan kepada pengguna, yaitu,

1. Pengumpulan data

Data pengembangan ternak didapatkan dari peternakan CV. Wahyu Utama di Tuban. Data tersebut adalah data dari 10 hewan ternak yang sedang melalui tahap penggemukan pada peternakan tersebut. Data yang didapatkan berupa foto, usia, warna kulit dan berat badan.

2. Pra-pemrosesan data

Setelah data dikumpulkan, dilakukan proses pengubahan data dari bentuk fisik menjadi bentuk digital melalui aplikasi.

3. Pengunggahan data ke basis data

Data digital dalam aplikasi telah siap untuk diunggah dalam basis data Firebase.

##### **4.1.2 Pembacaan Data**

Dalam Firebase pembacaan data dilakukan melalui *request* melalui Firebase API yang telah tersedia. Selanjutnya Firebase akan memberikan respon berupa *JSON* objek sesuai dengan lokasi yang ingin dibaca. Untuk mengambil data *JSON* dibutuhkan sebuah *class* sebagai inang untuk menguraikan *JSON* menjadi java objek. *Class* haruslah memiliki dua komponen utama yakni atribut dan *constructor* kosong. Selain itu kedua komponen tersebut dapat ditambahkan dengan *constructor* atau method lainnya.

### 4.1.3 Pemberitahuan

Proses pemberitahuan dilakukan dengan menggunakan *firebase cloud function*. Pemberitahuan akan dikirimkan kepada pengguna ketika terdapat pemicu dari pengguna pada *real time database*. Ketika pengguna menambahkan data ternak baru maupun ketika pengguna menambahkan perbaruan data ternak baru, *firebase cloud function* akan mendeteksi perubahan tersebut sebagai pemicu, sehingga *firebase cloud messaging* mengirimkan pesan pemberitahuan kepada *subscriber* data dalam hal ini adalah pengguna. Proses ini dapat dilihat pada gambar . Algoritma proses pengiriman pemberitahuan ini disajikan pada kode 1.

---

**Kode 1:** Pemberitahuan

---

```
1: function SENDNOTIFIKASITERNAKBARU(dBref.onWrite)      ▷
   dBref = path data ternak tersimpan
2:   pengguna = get ID pengguna
3:   payload = inisialisasi pesan
4:   if data == ada then
|     return
|     end
5:   kirimPesan(pengguna, payload)
6:   if sukses then
|     print "Pesan berhasil dikirim"
|     end
7:   else
|     print "Terjadi kesalahan" + kode error
|     end
8: end function
```

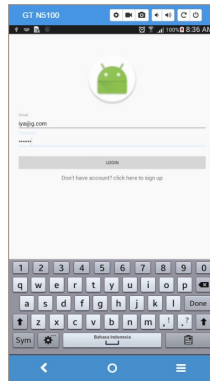
---

## 4.2 Hasil

Pada bagian ini dijelaskan mengenai hasil yang didapat dalam pengerjaan tugas akhir ini.

### 4.2.1 Hasil Antarmuka Aplikasi

Berikut merupakan hasil tampilan antarmuka aplikasi yang telah dibuat.



(b)

(a)

**Gambar 4.1:** Tampilan (a) *Login* dan (b) *Register*

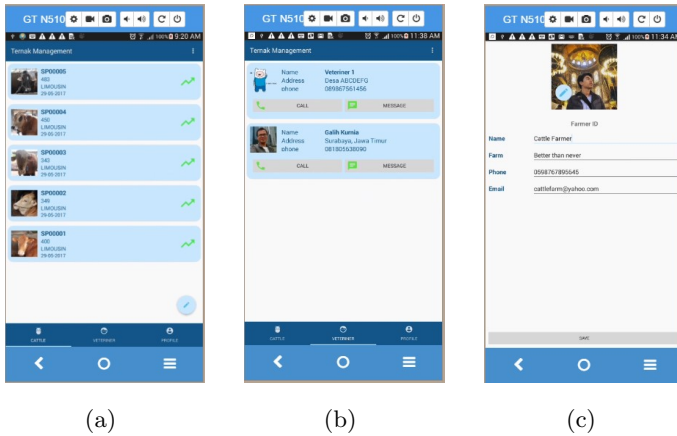
## 1. Tampilan Antarmuka *Login* dan *Register*

Halaman *Login* tampak seperti pada gambar 4.1a. Halaman ini akan muncul apabila pengguna tidak dalam kondisi *login* ke aplikasi. Namun apabila pengguna telah melakukan proses *login* dan belum melakukan proses *logout* dari aplikasi pengguna akan dibawa langsung menuju halaman utama. Pada halaman ini terdapat logo aplikasi, kolom *edit text* email, kolom *edit text* kata sandi, sebuah tombol *login* dan sebuah tombol *register*.

Halaman *Register* adalah halaman untuk pengguna melakukan registrasi apabila pengguna belum memiliki akun. Halaman ini dapat diakses dengan menekan tulisan yang berada di bawah tombol "Login" pada halaman *Login*. Pada halaman *Register* pengguna diwajibkan untuk mengisi semua kolom yang kosong. Pada kolom kata sandi pengguna diharuskan mengisi kata sandi dengan ketentuan minimal enam karakter. Halaman *Register* tampak seperti tampak pada gambar 4.1b.

## 2. Tampilan Antarmuka Utama

Halaman utama aplikasi ini terdiri dari tiga *fragment* yakni *fragment* ternak, *fragment* veteriner dan *fragment* pro-

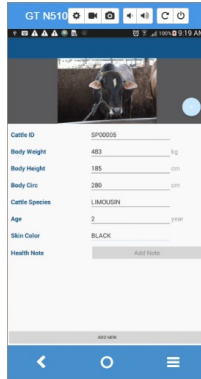


**Gambar 4.2:** Tampilan (a) *Fragment Ternak*, (b) *Fragment Veteriner* dan (c) *Fragment Profil*

fil.*Fragment ternak* dapat dilihat pada gambar 4.2a, *fragment* ini menampilkan daftar ternak yang dikelola oleh pengguna. Pada daftar ternak terdapat informasi ternak yang terbaru seperti gambar terakhir, berat badan terakhir, jenis ternak, ID ternak dan tanggal terakhir ternak diperbaharui.

*Fragment Veteriner* merupakan halaman yang berisi daftar dokter hewan yang dapat dihubungi oleh pengguna. Daftar ini memuat beberapa informasi singkat tentang dokter hewan seperti nama, alamat, nomer telpon dan foto. Pada halaman ini pengguna dapat menghubungi dokter hewan secara langsung melalui dua tombol yang tersemat pada setiap daftar yakni tombol *call* dan *message* untuk melakukan panggilan dan mengirim pesan. Halaman ini disajikan pada gambar 4.2b.

*Fragment Profil* berisi informasi tentang pengguna. Pada halaman ini pengguna dapat melakukan perubahan atau melengkapi data dirinya seperti foto, nama, nama peternakan yang dikelola, nomor telpon, alamat email dan jumlah ternak. Jumlah ternak pada halaman ini secara otomatis akan mengikuti jumlah ternak yang dikelola pengguna. Halaman dapat dilihat pada gambar 4.2c.



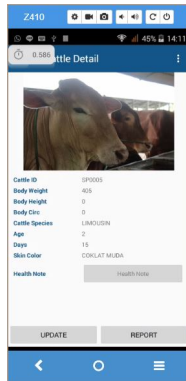
**Gambar 4.3:** Tampilan antarmuka Tambah Ternak

### 3. Tampilan Antarmuka Tambah Ternak

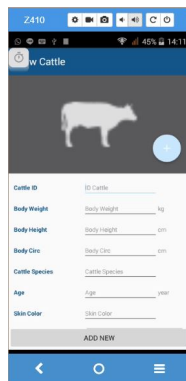
Halaman tambah ternak akan muncul ketika pengguna menekan *floating action button* tambah yang ada pada Halaman Utama Ternak. Pada halaman ini berisi formulir untuk menambahkan ternak baru. Formulir tersebut berisi *field* tambah gambar, *edit text* Id ternak, spesies ternak, berat badan ternak, tinggi badan ternak, lingkaran badan ternak, warna kulit dan umur ternak. Selain *field-field* tersebut terdapat juga tombol "Add New" yang berfungsi untuk mengirimkan ke basis data. Tampilan halaman ini dapat dilihat pada gambar 4.3. Ketika pengguna menekan tombol "Add New" dan data berhasil tersimpan pada basis data. Pengguna akan dibawa kembali pada tampilan detail ternak dengan data yang telah diperbaharui. Namun apabila data gagal tersimpan di basis data, pengguna akan mendapatkan *Toast* berisi pesan gagal menyimpan data.

### 4. Tampilan Antarmuka Detail Ternak

Halaman Detail Ternak adalah halaman yang memuat informasi detail tentang ternak. Halaman ini muncul ketika pengguna memilih salah satu ternak pada halaman utama. Halaman Detail Ternak dapat dilihat pada gambar 4.4. Pada halaman ini terdapat dua tombol navigasi untuk menuju ke



**Gambar 4.4:** Tampilan antarmuka Detail Ternak

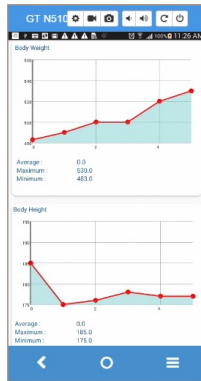


**Gambar 4.5:** Tampilan antarmuka Perbaharuan Ternak

halaman perbaharuan ternak dan halaman laporan ternak.

## 5. Tampilan Antarmuka Perbaharuan Ternak

Halaman Perbaharuan Ternak adalah halaman yang digunakan untuk menambahkan data terbaru suatu ternak. Pada halaman ini ditampilkan formulir untuk memasukkan data ternak seperti gambar atau foto, berat badan, tinggi badan, lingkaran badan dan warna kulit. Halaman ini dapat dilihat pada gambar 4.5



**Gambar 4.6:** Tampilan Laporan Ternak



**Gambar 4.7:** Tampilan pemberitahuan pembaharuan ternak

## 6. Tampilan Antarmuka Laporan Ternak

Halaman Laporan ternak adalah halaman yang menunjukkan laporan data ternak selama proses penggemukan. Data-data yang telah ditambahkan sebelumnya dikumpulkan dan dikelompokkan berdasarkan kategori dan kemudian disusun dalam grafik. Pada grafik juga dilengkapi dengan beberapa data tambahan seperti rata-rata pertumbuhan harian, pertumbuhan tertinggi dan pertumbuhan terendah. halam ini ditampilkan pada gambar 4.6.

### 4.2.2 Hasil Pemberitahuan

Pada tugas akhir ini pemberitahuan akan dikirim kepada pengguna dengan dua kondisi. Kondisi pertama adalah ketika pengguna menambahkan ternak baru dan tersimpan di basis data. Pemberitahuan ini berisi tentang informasi bahwa ternak telah berhasil disimpan pada basis data.

Kondisi kedua adalah ketika pengguna menambahkan data pembaharuan ternak. Pemberitahuan ini berisi tentang informasi pembaharuan ternak. Apabila berat badan ternak meningkat dari

data sebelumnya maka pesan pemberitahuan akan memberikan informasi pertumbuhan yang positif. Sebaliknya pemberitahuan akan berisi informasi pertumbuhan negatif apabila berat badan ternak lebih rendah dari data sebelumnya. Hasil pemberitahuan ini dapat dilihat pada gambar 4.7



## BAB 5

### PENGUJIAN DAN ANALISA

Pada Bab ini akan dijelaskan tentang pengujian dan analisa aplikasi Ternak Management yang dikembangkan. Pengujian dilakukan untuk memastikan aplikasi layak untuk digunakan pada kehidupan nyata sesuai dengan fungsinya.

#### 5.1 Metode Pengujian

Beberapa metode pengujian disiapkan untuk menguji kemampuan aplikasi yang dikembangkan, baik dari segi performa aplikasi ketika digunakan pada telepon pintar maupun dari segi komunikasi data dengan *server*. Metode pengujian tersebut adalah sebagai berikut,

1. *User Acceptance Test* (UAT)

*User Acceptance Test* (UAT) merupakan metode pengujian akhir terhadap suatu program, aplikasi atau perangkat lunak. Pengujian ini dilakukan untuk memastikan agar perangkat lunak yang dikembangkan mampu menangani tugas yang dibutuhkan pada skenario di kehidupan nyata sesuai spesifikasinya. Metode ini pada umumnya adalah pengujian tahap akhir yang dilakukan sebelum sebuah perangkat lunak diluncurkan ke pasar atau digunakan oleh pengguna.

2. Waktu Eksekusi

Metode waktu eksekusi adalah pengujian untuk mengetahui dan membandingkan kecepatan eksekusi perangkat lunak yang dikembangkan. Dari pengujian ini akan dapat dilihat efisiensi dari aplikasi saat digunakan di dunia nyata. Metode ini dibagi menjadi dua sub pengujian yaitu,

- (a) Waktu Eksekusi Unggah dan Unduh Data

Bagian ini akan menguji waktu unggah dan unduh data dari aplikasi kepada *server* dan sebaliknya. Pengujian ini dilakukan dengan mengambil *sample* data sebanyak dua puluh kali pada jaringan 3G dan 4G. Pengujian ini

dilakukan untuk mengetahui tingkat efisiensi komunikasi data pada aplikasi pada jaringan yang berbeda.

(b) Waktu Eksekusi Secara Bersamaan

Bagian ini akan menguji waktu eksekusi aplikasi pada beberapa perangkat secara bersamaan. Pengujian ini dilakukan untuk mengetahui efektifitas dan efisiensi basis data apabila diakses secara bersamaan.

## 5.2 *User Acceptance Test (UAT)*

Pengujian menggunakan UAT bertujuan untuk mengetahui bahwa aplikasi yang dikembangkan berfungsi sesuai dengan spesifikasi yang dibutuhkan pada kehidupan nyata. UAT berisi daftar uji coba yang diturunkan pada beberapa skenario pengujian. Skenario-skenario pengujian kemudian dibandingkan dengan luaran yang diinginkan oleh pengembang.

Hasil pengujian yang diperoleh adalah sebagai berikut,

1. Antar muka Login

Pada antar muka login, bagian yang diuji adalah kolom masukan untuk *username* dan *password*. Skenario pengujian dibagi menjadi dua yakni skenario berhasil dan skenario gagal. Keluaran yang diinginkan pada skenario berhasil adalah pengguna akan memasuki antar muka menu utama. Sedangkan skenario gagal memiliki keluaran notifikasi adanya kesalahan *username* atau *password*. Hasil yang didapatkan dari keduanya adalah sesuai dengan keluaran yang diharapkan.

2. Antar muka Register

Pada antar muka Register, bagian yang diuji adalah proses pembuatan akun dan batas minimal *password*. Skenario pengujian yang dilakukan untuk proses pembuatan akun adalah dengan melengkapi semua kolom masukan yang ada pada antar muka. Keluaran yang diinginkan adalah pengguna akan dibawa ke antar muka menu utama. Skenario pengujian batas *password* adalah dengan memasukkan *password* dengan ukuran karakter kurang dari enam. Keluaran yang diinginkan dari skenario ini adalah munculnya notifikasi bahwa *password* yang

dimasukkan harus lebih dari dan atau sama dengan enam karakter.

### 3. Antar muka Menu Utama-Daftar Ternak

Pada antar muka ini, bagian yang diuji adalah proses memasukkan data pada formulir ternak baru. Skenario pengujian yang dilakukan yakni skenario berhasil dengan mengisi seluruh kolom dan skenario gagal dengan mengosongi sebagian kolom. Luaran yang diinginkan pada skenario sukses adalah data akan muncul pada antar muka daftar ternak, sedangkan untuk skenario gagal adalah muncul notifikasi terjadinya kesalahan.

### 4. Antar muka Menu Utama - Daftar Dokter Hewan

Pada antar muka ini, bagian yang diuji adalah fitur untuk melakukan kontak kepada dokter hewan melalui aplikasi. Skenario pengujian yang dilakukan yakni melakukan panggilan dengan tombol *call* dan . Keluaran yang diinginkan adalah pengguna masuk ke antar muka panggilan dengan nomor telepon dokter hewan telah tertera. Skenario selanjutnya adalah mengirimkan pesan singkat dengan tombol *message* pada *item* dokter hewan. Keluaran dari skenario ini adalah pengguna akan masuk ke antar muka perpesanan dengan nomor telepon dokter hewan telah tertera.

### 5. Antar muka Menu Utama - Profil Pengguna

Pada antar muka ini, bagian yang diuji adalah penerimaan data profil pengguna dari *server* dan ditampilkan pada antar muka pengguna. Skenario pengujian dibagi menjadi skenario berhasil yakni dengan melakukan perubahan data pengguna dan skenario gagal yakni ketika data tidak berhasil tersimpan pada server. Keluaran yang diinginkan pada skenario berhasil adalah terlihat perubahan data pada profil pengguna. Sementara keluaran yang diinginkan ketika skenario gagal adalah muncul notifikasi terjadi kesalahan berupa *Toast* pada antar muka pengguna.

### 6. Antar muka Detail Ternak

Pada antar muka detail ternak, bagian yang diujikan adalah antar muka dapat menampilkan data terbaru ternak. Skenario yang dilakukan adalah pengguna memilih salah satu ternak pada antar muka daftar ternak. Keluaran yang diinginkan adalah munculnya semua data terbaru ternak pada antar muka pengguna.

#### 7. Antar muka Laporan Ternak

Pada antar muka laporan ternak, bagian yang diujikan adalah menampilkan akumulasi data laporan ternak berupa data detail, grafik dan analisis grafik (minimal, maksimal dan rata-rata) perkembangan ternak. Skenario pengujiannya yakni pengguna menekan tombol *report* pada antar muka Detail Ternak yang memunculkan antar muka Laporan Ternak. Keluaran yang diinginkan adalah semua data laporan ternak berhasil ditampilkan pada tempatnya seperti foto - foto, data detail dan data grafik serta hasil analisa grafik.

#### 8. Fitur *Logout*

Fitur logout akan memunculkan antar muka *Login* ketika dijalankan. Skenario pengujian yang dilakukan adalah pengguna melakukan proses *Logout* dengan menekan tombol *Logout*. Keluaran yang diinginkan adalah pengguna akan dibawa kembali pada antar muka *Login*.

Hasil pengujian dapat dilihat pada tabel 5.1. Dari tabel 5.1 dapat dilihat bahwa setiap skenario pengujian menghasilkan hasil yang sesuai seperti yang diinginkan.

**Tabel 5.1:** Hasil *User Acceptance Test* (UAT)

No	Antar Muka	Uji Coba	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Menu Login	Input <i>username</i> dan <i>password</i>	Memasukkan <i>username</i> dan <i>password</i> kemudian menekan tombol login	Dapat masuk kedalam mainmenu apabila <i>username</i> dan <i>password</i> benar.	Berhasil
2			Memasukkan <i>username</i> dan <i>password</i> yang salah kemudian menekan tombol login	Muncul notifikasi apabila <i>username</i> dan <i>password</i> salah.	Berhasil
3	Menu Registrasi	Proses Pembuatan akun pengguna	Memasukkan data yang dibutuhkan	Dapat masuk kedalam mainmenu	Berhasil
4		<i>Password</i> minimal yang dibutuhkan adalah enam karakter	Memasukkan password kurang dari enam karakter	Muncul notifikasi terjadi kesalahan	Berhasil
5	Menu Utama - Daftar Ternak	Menambahkan data ternak baru	Mengisi seluruh data yang dibutuhkan pada formulir	Data akan muncul pada halaman daftar ternak apabila seluruh daftar terisi	Berhasil
6			Mengosongi sebagian kolom pada formulir	Muncul notifikasi terjadi kesalahan	Berhasil
7	Menu Utama - Daftar Dokter Hewan	Dapat menghubungi kontak dokter hewan yang ada	Melakukan panggilan menggunakan tombol panggil	Ketika tombol ditekan, halaman panggilan akan muncul berikut nomor telepon dokter hewan sudah terisi	Berhasil
8			Mengirim pesan singkat menggunakan tombol pesan	Ketika tombol ditekan, halaman SMS akan muncul berikut nomor telepon dokter hewan sudah terisi	Berhasil
9	Menu Utama - Profil Pengguna	Menerima data detail pengguna dan ditampilkan pada halaman Profil Pengguna	Melakukan perubahan data profil dengan tombol edit	Profil pengguna berganti sesuai dengan data terbaru dari pengguna	Berhasil
10				Muncul notifikasi terjadi kesalahan ketika gagal melakukan edit profil	Berhasil
11	Menu Detail Ternak	Menampilkan data ternak terbaru pada halaman detail ternak	Pengguna memilih salah satu ternak pada daftar ternak dan memunculkan halaman detail ternak yang berisi data ternak	Data detail ternak terbaru muncul pada halaman detail ternak	Berhasil
12	Menu Laporan Ternak	Menampilkan data akumulasi laporan ternak berupa data detail dan grafik serta analisa data nilai minimal, nilai tertinggi dan rata-rata perkembangan ternak	Pengguna menekan tombol Report dan memunculkan halaman Laporan Ternak	Data laporan ternak berhasil ditampilkan pada halaman Laporan Ternak yakni foto-foto ternak, grafik perkembangan ternak dan analisisnya	Berhasil
13	Menu Logout	Muncul halaman login setelah pengguna menekan menu logout	Pengguna menekan tombol logout	Muncul halaman login setelah pengguna menekan menu logout	Berhasil

## 5.3 Waktu Eksekusi

Waktu eksekusi adalah waktu yang diperlukan untuk menyelesaikan sebuah perintah atau aktifitas. Pengujian ini dibagi menjadi dua bagian. Bagian pertama adalah pengujian aplikasi terhadap jaringan. Bagian kedua adalah pengujian aplikasi terhadap telepon pintar.

### 5.3.1 Waktu Eksekusi Unggah dan Unduh Data

Pengujian ini lebih menekankan pada waktu yang dibutuhkan aplikasi untuk melakukan unggah dan unduh data dari *server*. pengujian dilakukan pada tiga jaringan yang berbeda yakni 3G dan 4G. Hal ini dilakukan untuk mengetahui efektifitas aplikasi jika digunakan pada jaringan yang berbeda. Pengujian dilakukan sebanyak dua puluh kali percobaan pada setiap jaringan untuk mendapatkan data yang bervariasi. Pengujian untuk kategori ini sendiri terdapat beberapa sub uji yakni waktu eksekusi upload data, waktu eksekusi *authentication* pengguna dan waktu eksekusi pengunduhan data.

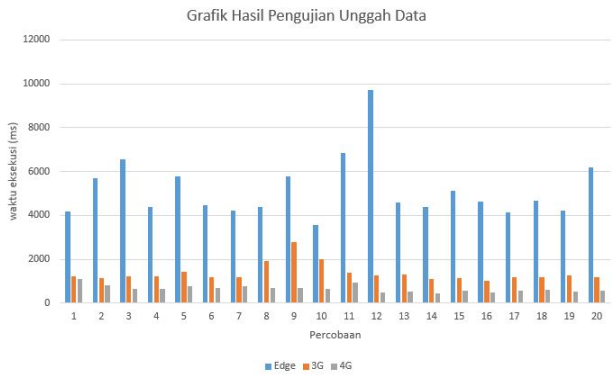
Waktu eksekusi upload data dilakukan pada saat pengguna mengunggah data detail ternak yang disertai file berupa foto dan data ternak lain. Hasil pengujian ini dapat dilihat pada tabel 5.2. Pada tabel 5.2 didapatkan rata-rata hasil pengujian pada jaringan 4G adalah 645 milidetik. Sedangkan pada jaringan 3G rata-rata hasil pengujian adalah 1363,25 milidetik. Grafik pengujian ini dapat dilihat pada grafik 5.1.

Waktu eksekusi *authentication* pengguna adalah waktu yang dibutuhkan aplikasi untuk melakukan proses pengenalan terhadap pengguna ketika *login* kedalam aplikasi. Hasil pengujian ini dapat dilihat pada grafik 5.2. Pada grafik 5.2 didapatkan rata-rata hasil pengujian pada jaringan 4G adalah 78.5 milidetik. Sedangkan pada jaringan 3G rata-rata hasil pengujian adalah 78.75 milidetik. Sedangkan pada jaringan Edge rata-rata hasil pengujian adalah 101 milidetik. Dari pengujian ini terlihat waktu eksekusi untuk proses *authentication* pengguna pada ketiga jaringan tidak memiliki perbedaan waktu yang signifikan, terutama pada jaringan 3G dan 4G yang relatif sama, sedangkan pada jaringan Edge sedikit lebih lambat.

Waktu eksekusi pengunduhan data pada pengujian kali ini

**Tabel 5.2:** Hasil pengujian waktu eksekusi unggah data

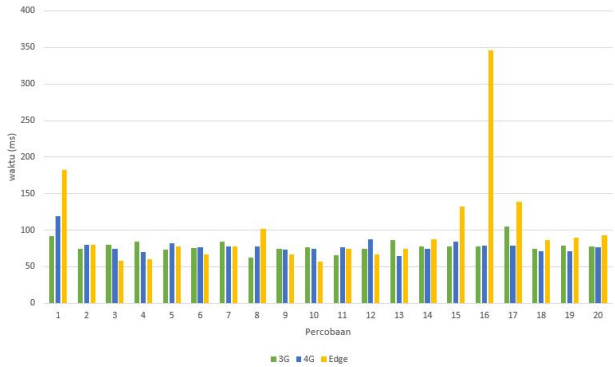
No	Waktu (ms)		
	3G	4G	Edge
1	1209	1099	4185
2	1135	795	5678
3	1219	659	6579
4	1243	632	4390
5	1424	777	5788
6	1189	693	4480
7	1197	754	4230
8	1905	693	4379
9	2763	681	5780
10	2022	629	3575
11	1374	925	6834
12	1248	465	9738
13	1298	539	4579
14	1094	455	4385
15	1126	565	5123
16	1019	480	4638
17	1186	547	4156
18	1168	593	4678
19	1278	542	4238
20	1168	557	6178
Rata-rata	1363.25	654	5180.55



**Gambar 5.1:** Grafik hasil pengujian waktu eksekusi unggah data

**Tabel 5.3:** Tabel hasil pengujian waktu eksekusi *authentication* data

No	Waktu (ms)		
	3G	4G	Edge
1	92	119	182
2	75	80	80
3	80	74	58
4	84	70	60
5	73	82	78
6	76	77	67
7	84	78	78
8	62	78	102
9	75	73	67
10	77	74	57
11	66	77	75
12	74	88	67
13	86	65	75
14	78	74	88
15	78	84	132
16	78	79	346
17	105	79	139
18	75	71	86
19	79	71	90
20	78	77	93
Rata-rata	78.75	78.5	101



**Gambar 5.2:** Grafik hasil pengujian waktu eksekusi *authentication* pengguna



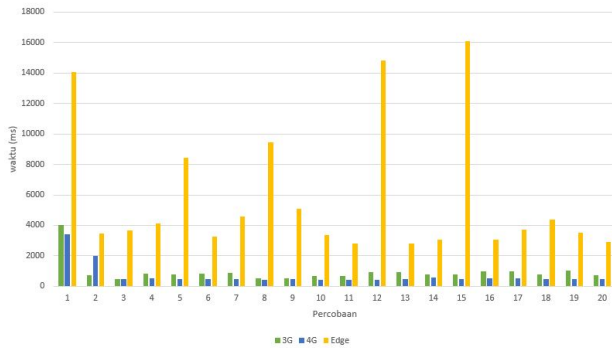
**Tabel 5.4:** Tabel hasil pengujian waktu eksekusi unduh data

No	Waktu (ms)		
	3G	4G	Edge
1	4033	3427	14083
2	725	1977	3441
3	492	449	3675
4	808	510	4141
5	781	461	8434
6	848	447	3275
7	868	451	4563
8	520	439	9436
9	514	459	5076
10	694	442	3349
11	660	430	2797
12	929	433	14810
13	951	470	2818
14	774	564	3076
15	801	458	16081
16	993	513	3042
17	989	502	3694
18	768	457	4394
19	1037	458	3534
20	745	477	2901
Rata-rata	946.5	691.2	5831

adalah waktu yang dibutuhkan aplikasi untuk memuat daftar data ternak pada antar muka utama. Hasil pengujian dapat dilihat pada grafik 5.3. Dari pengujian yang dilakukan didapatkan rata-rata hasil pengujian pada jaringan 4G adalah 691.2 milidetik. Sedangkan pada jaringan 3G rata-rata hasil pengujian adalah 946.5 milidetik. Sedangkan pada jaringan Edge rata-rata hasil pengujian adalah 5831 milidetik. Dari pengujian ini dapat dilihat bahwa waktu eksekusi untuk proses pengunduhan data 4G adalah yang tercepat disusul dengan jaringan 3G yang tidak terlalu jauh berbeda sedangkan pada jaringan Edge sangat jauh berbeda.

### 5.3.2 Waktu Eksekusi Secara Bersamaan

Bagian pengujian ini lebih menekankan pada waktu yang dibutuhkan *server* untuk diakses dari beberapa perangkat secara ber-



**Gambar 5.3:** Grafik hasil pengujian waktu eksekusi pengunduhan data

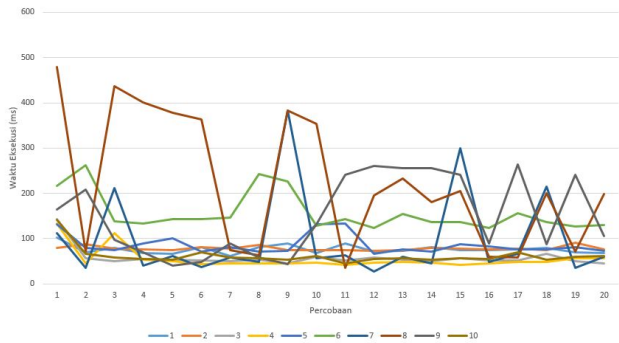
samaan. Pengujian ini dilakukan dengan menjalankan aplikasi pada sepuluh telepon pintar yang berbeda. Pengujian dilakukan untuk proses *authentication* pengguna dan mengunduh data.

Pada Tabel 5.5 ditunjukkan hasil pengujian eksekusi secara bersamaan pada proses *authentication* pengguna. Dari pengujian tersebut didapatkan rata-rata waktu tercepat adalah 62.1 milidetik sedangkan rata-rata waktu eksekusi terlama adalah 222.15 milidetik. Dari keseluruhan percobaan ini juga didapatkan bahwa waktu eksekusi secara bersamaan untuk proses *authentication* pengguna kurang dari satu detik dan tergolong cepat meskipun diakses secara bersamaan. Grafik hasil pengujian dapat dilihat pada grafik 5.4

Pada Tabel 5.6 ditunjukkan hasil pengujian eksekusi secara bersamaan pada proses mengunduh data daftar ternak. Dari pengujian tersebut didapatkan rata-rata waktu tercepat adalah 534.9 milidetik sedangkan rata-rata waktu eksekusi terlama adalah 2073.75 milidetik. Hal ini menunjukkan bahwa proses mengunduh data daftar ternak tergolong cepat ketika diakses secara bersamaan. Grafik hasil pengujian dapat dilihat pada grafik 5.5

**Tabel 5.5:** Hasil pengujian waktu eksekusi *authentication* data secara bersamaan

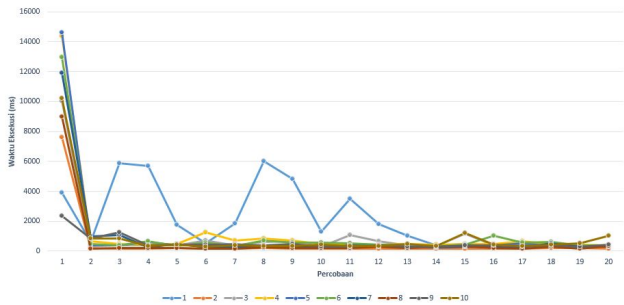
No	Waktu Eksekusi pada Perangkat (milidetik)									
	1	2	3	4	5	6	7	8	9	10
1	102	80	143	134	131	217	113	479	165	142
2	70	87	57	46	79	262	35	72	209	66
3	80	78	51	112	75	139	211	437	98	58
4	68	76	55	54	89	134	40	401	69	55
5	67	74	53	50	101	144	62	378	40	54
6	81	82	52	43	71	144	37	363	48	70
7	61	78	51	46	81	147	59	74	90	58
8	81	86	53	45	72	242	49	63	59	56
9	89	75	46	45	73	227	383	383	44	54
10	68	74	60	47	131	129	56	354	132	62
11	89	74	52	42	134	144	64	36	241	46
12	71	73	59	47	67	123	28	196	260	55
13	73	75	53	49	77	154	60	233	255	57
14	81	80	51	47	71	137	45	181	255	54
15	75	78	56	42	88	137	299	205	241	56
16	75	77	52	45	83	124	49	60	89	55
17	77	78	52	48	76	156	66	59	264	69
18	79	75	66	48	77	137	215	200	87	54
19	70	91	51	57	81	127	36	71	241	60
20	68	76	45	56	73	130	60	198	105	61
Rata-rata	76.25	78.35	57.9	55.15	86.5	157.7	98.35	222.15	149.6	62.1



**Gambar 5.4:** Grafik hasil pengujian waktu eksekusi *authentication* data secara bersamaan

**Tabel 5.6:** Hasil pengujian waktu eksekusi unduh data secara bersamaan

No	Waktu Eksekusi pada Perangkat (milidetik)									
	1	2	3	4	5	6	7	8	9	10
1	3906	7633	10030	14401	14596	12974	11923	8985	2366	10234
2	616	185	358	682	393	477	979	175	836	828
3	5898	156	369	461	383	385	1061	186	1262	825
4	5707	165	399	451	626	675	266	183	366	354
5	1765	189	394	471	372	355	485	185	432	413
6	509	179	707	1234	393	578	269	168	414	291
7	1838	150	368	711	476	358	265	177	432	377
8	5997	198	636	856	300	724	265	252	401	328
9	4838	164	653	693	300	573	254	191	486	344
10	1306	175	303	462	323	550	257	201	303	426
11	3478	161	1075	473	286	543	377	217	286	352
12	1789	160	679	447	289	436	325	276	392	345
13	1005	167	368	465	290	423	287	211	283	497
14	382	157	353	427	262	345	226	308	283	356
15	491	171	1162	461	288	448	290	331	372	1230
16	389	163	369	477	385	1024	257	190	392	447
17	404	175	377	677	509	563	233	179	293	351
18	385	185	296	460	332	631	530	306	383	432
19	379	196	290	446	295	390	308	178	303	526
20	393	165	351	395	314	387	310	326	413	1040
Average	2073.75	544.7	976.85	1257.5	1070.6	1141.95	958.35	661.25	534.9	999.8



**Gambar 5.5:** Grafik hasil pengujian waktu eksekusi pengunduhan data secara bersamaan

## **BAB 6**

### **PENUTUP**

#### **6.1 Kesimpulan**

Pada tugas ini telah dikembangkan aplikasi bergerak berbasis komputasi awan sebagai alat untuk membantu kegiatan kerja pemantauan perkembangan ternak pada peternakan. Fitur-fitur yang telah dirancang dan diterapkan pada aplikasi telah berfungsi sesuai dengan tujuan dengan dibuktikan pada pengujian UAT.

Waktu eksekusi yang dibutuhkan aplikasi untuk melakukan beberapa perintah bergantung pada jaringan seluler yang digunakan pada telepon pintar. Pada jaringan 4G aplikasi mengunggah data dengan rata waktu eksekusi 645 milidetik, jaringan 3G aplikasi mengunggah data dengan rata waktu eksekusi 1363,25 milidetik dan pada jaringan Edge aplikasi mengunggah data dengan rata waktu eksekusi 5180.55 milidetik. Untuk proses pengunduhan data waktu eksekusi yang dibutuhkan aplikasi pada jaringan 4G adalah 691.2 milidetik, 946.5 milidetik pada jaringan 3G dan 5831 milidetik pada jaringan Edge. Dari pengujian ini dapat disimpulkan bahwa jaringan Edge menghasilkan waktu paling lambat sedangkan waktu tercepat pada jaringan 4G.

Sementara pada pengujian secara bersamaan waktu eksekusi untuk proses pengunduhan data rata-rata tercepat adalah 534.9 milidetik dan waktu terlambat adalah 2073.75 milidetik. Dari pengujian ini dapat disimpulkan bahwa aplikasi mampu bekerja dengan baik ketika diakses secara bersamaan.

#### **6.2 Penelitian Selanjutnya**

Pada penelitian selanjutnya akan dilakukan penambahan fitur-fitur baru pada aplikasi untuk lebih membantu proses pemantauan perkembangan ternak bagi para peternak. Fitur yang dapat ditambahkan seperti pemantauan kesehatan ternak, pemindai kode baris dan menghubungkan aplikasi dengan perangkat *Internet of Thing (IoT)*. Selain itu dapat dilakukan perbaikan pada tampilan agar lebih *responsive* untuk ukuran layar telepon pintar yang bervariasi.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] R. Suryani dkk, Outlook Komoditas Pertanian Subsektor Peternakan Daging Sapi. Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian, 2015. (Dikutip pada halaman 1).
- [2] G. M. Peter dan, Tim, “The nist definition of cloud computing,” National Institute of Standards and Technology, vol. 53, no. 6, p. 50, 2009. (Dikutip pada halaman 5).
- [3] J. Strickland, “How cloud computing works.” <http://computer.howstuffworks.com/cloud-computing/cloud-computing1.htm>, 2008. diakses pada 7 Juni 2017. (Dikutip pada halaman 5).
- [4] K. J Yashpalsinh dan M “Cloud computing-concepts, architecture and challenges,” in Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on, pp. 877–880, IEEE, 2012. (Dikutip pada halaman 5, 6).
- [5] M. Rayhan, “Berbagi pengalaman menggunakan firebase dalam membantu pengembangan android apps.” [https://medium.com/@emd\\_ray/berbagi-pengalaman-menggunakan-firebase-dalam-membantu-2016](https://medium.com/@emd_ray/berbagi-pengalaman-menggunakan-firebase-dalam-membantu-2016). diakses pada 5 Mei 2017. (Dikutip pada halaman 7).
- [6] M. A. Mohammed dkk, “Mobile-based medical health application-medi-chat app,” vol. 6, 2017. (Dikutip pada halaman 7).
- [7] F. Documentation, “Firebase authentication.” <https://firebase.google.com/docs/auth/>, 2017. diakses pada 7 Mei 2017. (Dikutip pada halaman 7).
- [8] F. Documentation, “Firebase realtime database.” <https://firebase.google.com/docs/database/>, 2016. diakses pada 7 Mei 2017. (Dikutip pada halaman 8, 9, 10, 11).
- [9] F. Documentation, “Firebase storage.” <https://firebase.google.com/docs/storage/>, 2016. diakses pada 7 Mei 2017. (Dikutip pada halaman 9).

- [10] F. Documentation, “Firebase storage.” <https://firebase.google.com/docs/functions/>, 2017. diakses pada 7 Mei 2017. (Dikutip pada halaman 9).
- [11] A. Narayanan, “Denormalizing your data is normal.” <https://firebase.googleblog.com/2013/04/denormalizing-your-data-is-normal.html>, 2013. diakses pada 7 Mei 2017. (Dikutip pada halaman 11).



## BIOGRAFI PENULIS



Nailul Fadloil, lahir pada 2 Juni 1995 di Demak, Jawa Tengah. Penulis lulus dari SMP Negeri 2 Demak pada tahun 2010 kemudian melanjutkan pendidikan ke SMA Negeri 1 Kudus dan lulus pada tahun 2013. Penulis kemudian melanjutkan pendidikan jenjang strata satu pada Departemen Teknik Komputer ITS Surabaya bidang studi Telematika. Saat di kuliah penulis aktif menjadi Asisten laboratorium B401 (Komputasi Digital) hingga saat ini dan pernah menjabat sebagai koordinator asisten Lab B401 periode

2015/2016. Penulis juga aktif dalam beberapa organisasi mahasiswa seperti BEM ITS, UKM CLICK dan beberapa kegiatan kepanitiaan lainnya. Selama kuliah penulis aktif dalam mengikuti ajang perlombaan seperti *Hackaton*, *Kompfest*, Karya Tulis Ilmiah dan Program Kreativitas Mahasiswa. Penulis sangat tertarik dengan segala hal yang berhubungan dengan Teknologi Informasi dan berencana untuk mendalami bidang Teknologi Informasi lain yang spesifik pada sistem cerdas, sisten terintegrasi dan *smartcity*.

*email* : nailulfadloil@gmail.com  
*phone* : +62 8180 - 6484 - 509

*Halaman ini sengaja dikosongkan*